AMENDMENT

Please amend the above-identified application as follows:

Amendments to the Claims:

1.   (Currently Amended)   A method for propagating an application wherein the application includes a plurality of components, said method comprising:

selecting a destination environment;

propagating the components from a source environment to the destination environment according to a set of rules; and

wherein the propagation of at least two of the components from the source environment to the destination environment is performed in parallel using multiple instances of a difference engine;

wherein the difference engine uses rules that are applied at different stages to determine the propagation of a component of the components, the stages include a first stage that applies a first rule if the component is ~~detected~~ deleted in the source environment, a second stage that applies a second rule if the component is ~~detected~~ deleted in the destination environment, a third stage that applies a third rule if the component is modified in the source environment, a fourth stage that applies a forth rule if the component is modified in the destination environment, a fifth stage that applies a fifth rule if the component is new in the source environment, and a sixth stage that applies a sixth rule if the component is new in the destination environment;

wherein an interface allows for the selection of any of the first, second, third, fourth, fifth and sixth rule.

2. (Original) The method of claim 1 wherein:

the application can be a web application.

3. (Original) The method of claim 1 wherein:

the plurality of components can include at least one of: binary files, J2EE

(Enterprise Java) applications, .Net applications, LDAP information, distributed objects,

libraries, configuration files, information in databases including database records, Java

Archives (JARs), XML (Extensible Markup Language) documents, and HTML

(Hypertext Markup Language) documents.

4. (Previously Presented) The method of claim 1 wherein:

the plurality of components is distributed on a plurality of source operating

environments.

5. (Original) The method of claim 1 wherein:

a rule in the set of rules can determine whether the source environment or the

destination environment take precedence.

6. (Previously Presented) The method of claim 1, further comprising:

providing a user interface; and

wherein the user interface initiates the propagation.

7.   (Original)   The method of claim 6 wherein:

the user interface provides a first user interface to allow a user to create one or more

rules in the set of rules.

8.   (Original)   The method of claim 6 wherein:

the user interface provides a first user interface to allow a user to preview the

changes that will take place in the destination environment.

9.   (Original)   The method of claim 1, further comprising:

providing a process interface to allow a process to initiate the propagation.

10.   (Original)   The method of claim 1 wherein:

the source and/or destination environment can include a plurality of computing

devices.

11.   (Currently Amended)   A computer-implemented system for propagating an

application wherein the application includes a plurality of components, said system

comprising:

a process interface operable to accept propagation requests;

a difference engine operable to propagate the components from a source

environment to a destination environment according to a set of rules;

threading model operable to instantiate instances of the difference engine; and

wherein the propagation of at least two of the components from the source

environment to the destination environment is performed in parallel using multiple

instances of the difference engine;

wherein the difference engine uses rules that are applied at different stages to

determine the propagation of a component of the components, the stages include a first

stage that applies a first rule if the component is ~~detected~~ <u>deleted</u> in the source

environment, a second stage that applies a second rule if the component is ~~detected~~

<u>deleted</u> in the destination environment, a third stage that applies a third rule if the

component is modified in the source environment, a fourth stage that applies a forth rule

if the component is modified in the destination environment, a fifth stage that applies a

fifth rule if the component is new in the source environment, and a sixth stage that

applies a sixth rule if the component is new in the destination environment<u>;</u>

<u>wherein an interface allows for the selection of any of the first, second, third,</u>

<u>fourth, fifth and sixth rule</u>.

12.   (Canceled)

13.   (Previously Presented)   The computer-implemented system of claim 11 wherein:

the application can be a web application.

14.   (Previously Presented)   The computer-implemented system of claim 11 wherein:

the plurality of components can include at least one of: binary files, J2EE

(Enterprise Java) applications, .Net applications, LDAP information, distributed objects,

libraries, configuration files, information in databases including database records, Java

Archives (JARs), XML (Extensible Markup Language) documents, and HTML

(Hypertext Markup Language) documents.

15. (Previously Presented) The computer-implemented system of claim 11 wherein:

the plurality of components can be distributed on a plurality of source operating

environments.

16. (Previously Presented) The computer-implemented system of claim 11 wherein:

a rule in the set of rules can determine whether the source environment or the

destination environment take precedence.

17. (Previously Presented) The computer-implemented system of claim 11, further

comprising:

a user interface; and

wherein the user interface initiates the propagation.

18. (Previously Presented) The computer-implemented system of claim 17 wherein:

the user interface provides a first user interface to allow a user to create one or more

rules in the set of rules.

19. (Previously Presented) The computer-implemented system of claim 17 wherein:

the user interface provides a first user interface to allow a user to preview the changes that will take place in the destination environment.

20. (Previously Presented) The computer-implemented system of claim 17 wherein:

the source and/or destination environment can include a plurality of computing devices.

21. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

select a destination environment;

propagate a plurality components of an application from a source environment to the destination environment according to a set of rules; and

wherein the propagation of at least two of the components from the source environment to the destination environment is performed in parallel using multiple instances of a difference engine;

wherein the difference engine uses rules that are applied at different stages to determine the propagation of a component of the components, the stages include a first stage that applies a first rule if the component is ~~detected~~ deleted in the source environment, a second stage that applies a second rule if the component is ~~detected~~ deleted in the destination environment, a third stage that applies a third rule if the component is modified in the source environment, a fourth stage that applies a forth rule if the component is modified in the destination environment, a fifth stage that applies a

fifth rule if the component is new in the source environment, and a sixth stage that

applies a sixth rule if the component is new in the destination environment;

<u>wherein an interface allows for the selection of any of the first, second, third,</u>

<u>fourth, fifth and sixth rule</u>.

22.   (Original)   The machine readable medium of claim 21 wherein:

the application can be a web application.

23.   (Original)   The machine readable medium of claim 21 wherein:

the plurality of components can include at least one of: binary files, J2EE

(Enterprise Java) applications, .Net applications, LDAP information, distributed objects,

libraries, configuration files, information in databases including database records, Java

Archives (JARs), XML (Extensible Markup Language) documents, and HTML

(Hypertext Markup Language) documents.

24.   (Original)   The machine readable medium of claim 21 wherein:

the plurality of components can be distributed on a plurality of source operating

environments.

25.   (Original)   The machine readable medium of claim 21 wherein:

a rule in the set of rules can determine whether the source environment or the

destination environment take precedence.

26. (Previously Presented) The machine readable medium of claim 21, further comprising instructions that when executed cause the system to:

provide a user interface; and

wherein the user interface initiates the propagation.

27. (Original)   The machine readable medium of claim 26 wherein:

the user interface provides a first user interface to allow a user to create one or more rules in the set of rules.

28. (Original)   The machine readable medium of claim 26 wherein:

the user interface provides a first user interface to allow a user to preview the changes that will take place in the destination environment.

29. (Original)   The machine readable medium of claim 21, further comprising instructions that when executed cause the system to:

provide a process interface to allow a process to initiate the propagation.

30. (Original)   The machine readable medium of claim 21 wherein:

the source and/or destination environment can include a plurality of computing devices.

31. (New) The method of claim 1, wherein:

the first rule specifies whether the difference engine should keep the component in the destination environment or to remove it.

32. (New)   The method of claim 1, wherein:

the second rule specifies whether the difference engine should copy the component to the destination environment.

33. (New)   The method of claim 1, wherein:

the third rule specifies whether the difference engine should overwrite the component in the destination environment with the component from the source environment.

34. (New)   The method of claim 1, wherein:

the fourth rule specifies whether the difference engine should overwrite the component in the destination environment with the component from the source environment.

35. (New)   The method of claim 1, wherein:

the fifth rule specifies whether the difference engine should copy the new component to the destination environment.

36. (New)   The method of claim 1, wherein:

the sixth rule specifies whether the difference engine should remove the new component from the destination environment.

37. (New)   The method of claim 1, wherein:

the first rule specifies whether the difference engine should keep the component in the destination environment or to remove it;

the second rule specifies whether the difference engine should copy the component to the destination environment;

the third rule specifies whether the difference engine should overwrite the component in the destination environment with the component from the source environment;

the fourth rule specifies whether the difference engine should overwrite the component in the destination environment with the component from the source environment;

the fifth rule specifies whether the difference engine should copy the new component to the destination environment; and

the sixth rule specifies whether the difference engine should remove the new component from the destination environment.

38. (New)   The method of claim 1, wherein:

at least one of the first, second, third, fourth, fifth and sixth rules indicate whether to present a display to a user for a decision.